

웹 프로그래밍

# 7장. CSS, 자바스크립트 입문하기

동아대학교 컴퓨터공학과

# 7.1 CSS

# CSS

## ■ 스타일시트란?

- 웹 문서의 출력될 외형 스타일
  - ▶ HTML 태그로는 세세한 부분까지 모두 다 지정하기에는 부족
  - ▶ 스타일시트를 이용하면 크기, 색상 등의 스타일을 일괄 적용
    - 글자간격, 문단간격, 위치 등 자세한 부분까지 제어
- 콘텐츠의 내용과 디자인의 분리가 가능
  - ▶ 웹문서에서 마크업 요소는 보다 내용의 구조에 치중
  - ▶ 디자인 요소는 별도로 작성

## ■ CSS(Cascading Style Sheet)

- 웹 문서용으로 개발한 스타일시트 언어

- 스타일 지정이 없는 문서

```
<html>
  <body>
    <h3>스타일시트</h3>
    <p>안녕? 나는 <strong>CSS</strong>란다~</p>
    <p>너 디자인 안목 좀 있냐?</p>
  </body>
</html>
```

### 스타일시트

안녕? 나는 CSS란다~  
너 디자인 안목 좀 있냐?

- 폰트 및 색상 지정

```
<html>
  <head>
    <style type="text/css">
      h3 { font-style:italic }
      p { font-size:20pt; color:red }
    </style>
  </head>
  <body>
    <h3>스타일시트</h3>
    <p>안녕? 나는 <strong>CSS</strong>란다~</p>
    <p>너 디자인 안목 좀 있냐?</p>
  </body>
</html>
```

### 스타일시트

안녕? 나는 CSS란다~  
너 디자인 안목 좀 있냐?

# CSS 선택자 종류 (4가지 中)

## 1. 태그 선택자 종류1

- 다중 태그는 콤마(,)로 구분, 다중 속성은 세미콜론(;)으로 구분

예) `h3, strong { color: red; font-style: italic }`

- 다중 속성값은 콤마로 나열 : 순서대로 가능한 속성값을 적용

예) `p { font: Palatino, Garamond, "Times New Roman"; font-size: small }`

- 선택자 조합

▶ 보다 구체적인 요소의 선택이 가능

예) `h3 strong { font-style: italic }` /\* h3에 속하는 strong 요소 \*/

`p strong { color: red }` /\* p에 속하는 strong 요소 \*/



※ 심포 있을 때랑 없을 때는 달라요~

# CSS 선택자 종류 (4가지 中)

## 2. 클래스 선택자 종류2

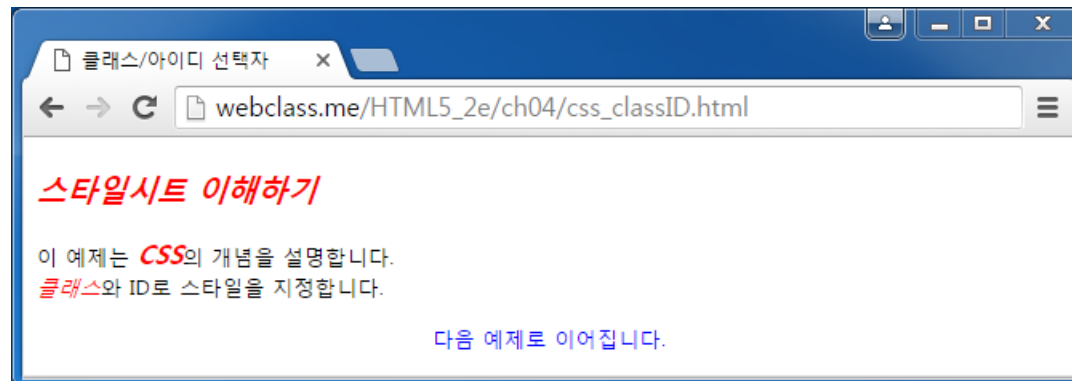
- 여러 태그에 특정 스타일을 공통으로 적용하고자 할 때
  - ▶ class 속성 이용 : <태그이름 **class="클래스이름"**> ... </태그>
- 클래스 선택자는 클래스이름 앞에 점(.)
  - ▶ **".클래스이름"** : 해당 클래스에 모두 적용  
**.red1** {color: red; font-style: italic; }
  - ▶ **"선택자.클래스이름"** : 특정 태그에서 해당 클래스만 지정  
**strong.red1** {font-size: 12pt }

## 3. 아이디 선택자 종류3

- 아이디는 html 문서 내에서 한 군데에서만 지정 가능  
<태그이름 **id="아이디이름"**> ... </태그>
- 해당 아이디로 설정된 태그에만 특정 스타일을 적용
  - ▶ 아이디 선택자는 아이디이름 앞에 샵(#)을 붙인다  
**#next** { color:blue; text-align:center}

# 1,2,3번 선택자 예제

```
<head>
  <style>
    p {font-size: 10pt}
    .red1 {color: red; font-style: italic; }    /* red1 클래스는 빨간색 이탤릭으로 */
    strong.red1 {font-size: 12pt }             /* strong 요소중 red1 클래스는 12pt 크기 */
    #next { color: blue; text-align: center}    /* next 아이디는 파란색 가운데 정렬 */
  </style>
</head>
<body>
  <h3 class="red1">스타일시트 이해하기</h3>
  <p>이 예제는 <strong class="red1">CSS</strong>의 개념을 설명합니다.
  <br><span class="red1">클래스</span>와 ID로 스타일을 지정합니다.</p>
  <p id="next">다음 예제로 이어집니다.</p>
</body>
```



## 종류4

# CSS 선택자 종류 (4가지)

## 4. 가상클래스(pseudo class) 선택자

- 요소 이름 다음 콜론(:) 뒤에 예약어
  - ▶ 요소를 선택할 수 있는 특별한 상태를 표현
  - ▶ 예, a:link는 링크를 의미, 방문한 링크는 a:visited로 표현
  - ▶ 예, :before 와 :after 는 content 속성으로 원하는 콘텐츠 추가

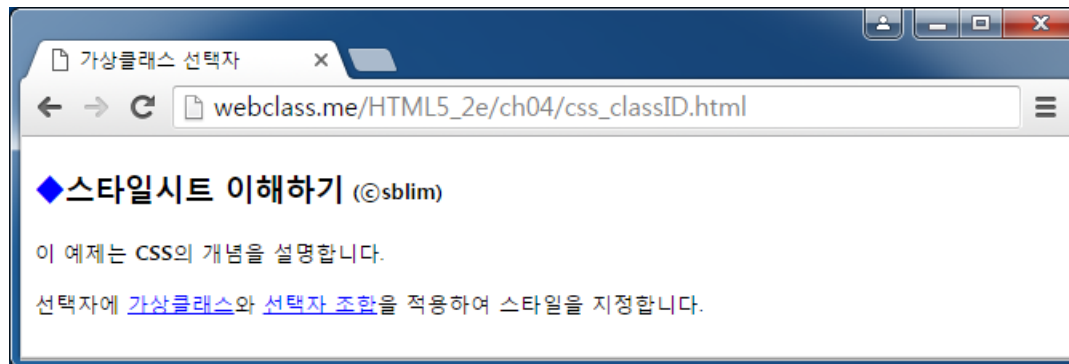
## ■ 대표적인 가상클래스 선택자

- 하이퍼링크 관련 :link :visited
- 마우스 관련 :active :hover :focus    마우스 관련은 안 외워도 됨
- 콘텐츠 삽입 :before :after



# 4번 선택자 예제

```
<head>
  <style>
    p {font-size:10pt}
    a:link { color: blue; }           /* a태그의 하이퍼링크 */
    a:visited { color: green; }       /* 방문한 a태그의 링크 */
    h3:before { content: "◆"; color: blue } /* h3요소의 앞에 파란색 ◆문자 삽입 */
    h3:after { content: " (©sblim)"; font-size:10pt } /* h3요소의 뒤에 콘텐츠 삽입 */
  </style>
</head>
<body>
  <h3><strong>스타일시트</strong> 이해하기</h3>
  <p>이 예제는 <strong>CSS</strong>의 개념을 설명합니다.</p>
  <p>선택자에 <a href="http://www.w3c.org">가상클래스</a>와
    <a href="http://www.donga.ac.kr">선택자 조합</a>을 적용하여 스타일을 지정합니다.</p>
</body>
```



# 색상

## ■ 색상의 표현 : RGB 혹은 RGBA 모델

- 화면의 각 점(픽셀)은 3바이트 혹은 4바이트
  - ▶ 각각 1 바이트씩 RGB(Red, Green, Blue) 색상값 : 0~255까지 표현
  - ▶ RGBA 모델의 경우 4번째 바이트는 투명색 표현 등 특수용도 사용

## ■ 색상 값의 표현

- 16진수 표현: #RRGGBB 예) #ff0000, #080800
- 10진수 표현 함수: rgb(R, G, B) 예) rgb(255, 0, 0), rgb(128, 128, 0)
- 백분율 표현 함수: rgb(R%, B%, G%) 예) rgb(50%, 50%, 0%),
- 키워드 표현:
  - ▶ 투명색은 transparent라는 키워드로 표현
  - ▶ 색상 이름 : red, lime, blue, yellow, aqua, fuchsia, silver, gray, ...

# 배경

## ■ 배경 관련 속성

- 각 요소의 영역에 배경색이나 이미지를 배경으로 지정
- 관련 속성
  - ▶ background-color
  - ▶ background-image : url(파일경로)
  - ▶ background-repeat
  - ~~▶ background-attachment : scroll, fixed~~
  - ▶ background-position

# 색상 및 배경 예제

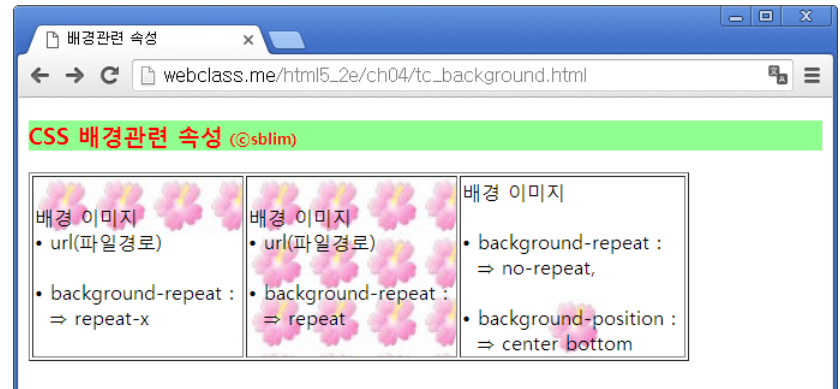
```
<head>
  <style>
    h3 { color: red; background-color: #90ff90 }
    h3:after { content: " (©sblim)"; font-size:10pt; }
    #first { background-image: url(flower.jpg); background-repeat:repeat-x; }
    #second { background-image: url(flower.jpg); }
    #third { background-image: url(flower.jpg); background-repeat:no-repeat;
              background-position: center bottom }
  </style>
</head>
<body>
  <h3> CSS 배경관련 속성</h3>
  <table border="1">
    <tr>
      <td id="first"> 배경 이미지<br> &bull; url(파일경로) <br><br>&bull;
        background-repeat : <br> &nbsp; => repeat-x </td>
      <td id="second"> 배경 이미지<br> &bull; url(파일경로) <br><br>&bull;
        background-repeat : <br> &nbsp; => repeat </td>
      <td id="third"> 배경 이미지 <br><br>&bull; background-repeat : <br>
        &nbsp; => no-repeat, <br><br>&bull; background-position : <br>
        &nbsp; => center bottom</td>
    </tr>
  </table>
</body>
```

원문자 등의 특수문자를 바로 입력해도 됨  
(바로 타이핑하기 힘들면, 아래아한글 등 다른 곳에서 복사해와도 됨)

4 는 세로

아무말 없으면 가로세로 모두 반복

&bull; 는 글머리 기호에 많이 쓰이는 동글이  
(안 외워도 됨)



# 스타일시트 선언 방법 (3가지 中)

## 1. 내부 스타일시트 선언

- HTML 문서의 <head>에서 <style> 태그를 이용하여 선언
- 주석은 /\* 와 \*/ 사이에 기입
  - ▶ <head>
  - ▶ <style> /\*CSS 스타일 선언\*/ </style>
  - ▶ </head>

```
<html>
<head>
  <style>
    h3 {color:red}           /* h3의 색상을 빨간색으로 */
    p {font-size:10pt}       /* p의 글자를 한 크기 작게 */
  </style>
</head>
<body>
  <h3>스타일시트 이해하기</h3>
  <p>이 예제는 <strong>CSS</strong>의 개념을
설명합니다.</p>
</body>
</html>
```

## 위치2

# 스타일시트 선언 방법 (3가지 中)

## 2. 외부 스타일시트 연결

- HTML 문서의 <head>에서 <link> 태그를 이용하여 연결

▶ <head>

<link rel="stylesheet" type="text/css" href="a.css" />

▶ </head>

생략가능

a.css

```
/* 외부 스타일시트 */  
h3 { color:red}  
p { font-size:10pt }
```

a.html

```
<html>  
  <head>  
    <link rel="stylesheet" type="text/css" href="a.css"/>  
  </head>  
  <body>  
    <h3>스타일시트 이해하기</h3>  
    <p>이 예제는 <strong>CSS</strong>의 개념을  
    설명합니다.</p>  
  </body>  
</html>
```

## 스타일시트 선언 방법 (3가지 中)

### 위치3

### 3. 인라인 스타일시트 삽입

- 모든 요소에서 사용할 수 있는 style 속성을 이용
- 해당 태그에만 특정한 스타일을 적용하고 싶을 때 사용

▶ <태그 style="CSS 속성선언">

```
<html>
  <head>
    <style>
      h3 {color:red; font-style:italic; }
      p {font-size:10pt}
    </style>
  </head>
  <body>
    <h3>스타일시트 이해하기</h3>
    <p>이 예제는 <strong style="font-style:italic; color:red;"> CSS</strong>의
      개념을 설명합니다.</p>
    <p>다음 예제로 이어집니다.</p>
  </body>
</html>
```

# 넓이, 높이 예제

```
<head>
  <style>
    td {width:300px; height:200px}
  </style>
</head>
<body>
  <table border=1>
    <tr><td></td><td></td></tr>
    <tr><td></td><td></td></tr>
  </table>
</body>
```

px는 고정 길이이므로, 웹화면 크기  
정해도, 테이블 크기는 변하지 않음

But in general you would use a different set of units for display on screen than for printing on paper. The following table gives the recommended use:

	Recommended	Occasional use	Not recommended
Screen	em, px, %	ex	pt, cm, mm, in, pc
Print	em, cm, mm, in, pt, pc, %	px, ex	

```
<head>
  <style>
    table {width:50%; height:50%}
  </style>
</head>
<body>
  <table border=1>
    <tr><td></td><td></td></tr>
    <tr><td></td><td></td></tr>
  </table>
</body>
```

%는 상대 길이이므로, 웹화면  
크기 조정하면, 테이블 크기도 변함

The relation between the absolute units is as follows: 1 in = 2.54cm = 25.4mm = 72pt = 6pc



CSS는 끝이 없음... 디자인에 끝이 없듯이...  
우리는 이 정도(CSS가 뭔지 아는 정도)만 함.

## 7.2 자바 스크립트

# 1. 자바스크립트란?

- 동적인 웹 문서 제작과 웹 응용프로그램 개발을 위한 사용자 인터페이스 개발을 위해서 필수적으로 사용됨

	자바스크립트	자바 언어
실행 방식	웹 브라우저에서 바로 자바스크립트 코드를 해석하고 바로 실행 ( <u>스크립트/인터프리터 기반 언어</u> )	자바 프로그램을 컴파일 후 변환된 object code를 자바가상머신에서 실행하는 방식 ( <u>컴파일 기반 언어</u> )
작성 형태	HTML 파일 내에 포함되어 작성됨	별도의 자바 프로그램 파일로 작성
변수형 선언 및 타입 검사	변수의 선언이 따로 필요 없으며 타입 검사도 매우 느슨함	변수의 선언이 필요하며 변수 타입의 검사가 매우 엄격함

## 2. 자바스크립트 실행 및 작성

- HTML 파일 내에 자바스크립트 코드를 만나면 웹 브라우저가 자체 인터프리터를 이용해 해석 및 실행
  - 자바스크립트를 실행하는 동안 오류가 발생하더라도 치명적 오류가 아니라면 기본적으로 웹 브라우저는 그 오류를 무시하고 진행
- 자바스크립트 코드는 HTML 파일 없이 웹브라우저에서 독립적으로 실행 될 수 없음
  - 반드시 HTML 파일 내에 포함되어 있어야 한다
- HTML 파일 내에 포함 시키는 두 가지 방식
  - 웹문서 내장 방식
  - 외부 파일 참조 방식

## (2.1) 웹문서 내장 방식

```
1 <!-- HTML documents... -->
2 ....
3 <script type = "text/javascript">
4     <!--
5         // 자바스크립트 코드
6     // -->
7 </script>
8 ....
9 <!-- HTML documents... -->
```

- 3: 자바스크립트 포함의 시작을 알리는 **<script>** 태그
- 4: 특수 주석 시작
- 5: 실제 자바스크립트 코드들이 위치하는 곳
- 6: 특수 주석 끝
- 7: 자바스크립트 포함의 끝을 알리는 클로징 태그

X

## (2.2) 외부 파일 참조 방식

- `<script>` 태그의 `src` 속성의 값으로 자바스크립트 파일의 경로를 지정

```
1 <!-- HTML documents... -->
2 ....
3 <script src="myscript.js">
4 </script>
5 ....
6 <!-- HTML documents... -->
```

src = source

- 자바스크립트 파일의 URL 경로 지정 가능

```
1 <!-- HTML documents... -->
2 ....
3 <script src="http://webclass.me/html5/javascript_example.js">
4 </script>
5 ....
6 <!-- HTML documents... -->
```

### 3. 자바스크립트 변수 **(var)** ( = variable )

- 변수를 사전에 **선언 없이 사용하는 것이 가능**
  - 전역 변수로 사용할 때는 미리 선언되어 있어야 함
  - 변수 타입을 고려하지 않고 선언해서 사용하면 되므로 편리
- 별도의 변수 타입이 없으며 **var** 타입 한가지만 제공됨
  - 대신, 변수에 실제로 어떤 값이 저장될 때 그 값에 따라 내부적으로 변수 타입이 정해진다

```
var index, name = "모바일 웹";  
var start = 0, end = 100.0;  
var message, condition, sender, receiver;
```

```
var a = "3";  
var b = 2;  
c = b + 3 + a;    // c값은 "53"이 됨, 더하기 후 문자열 붙이기 연산  
d = a + b;        // d값은 "3"+"2"="32", 문자열연산(Concatenation)
```

## 4. 타입 변환 예제

- 문자열 타입 → 숫자 타입    `parseInt()`, `parseFloat()`
- 숫자 타입 → 문자열타입    `toString()`

참고: JSP에서는  
`Integer.parseInt()` 옴

<script>

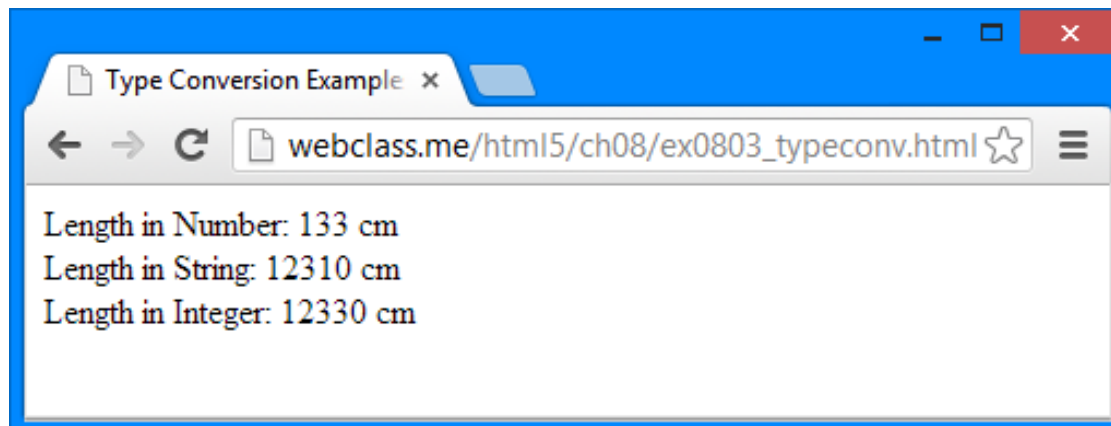
```
var length = 123, length_num, length_str;  
length_num = length + 10;
```

```
length_str = length.toString() + 10;  
document.write("Length in Number: " + length_num + " cm" + "<br />");  
document.write("Length in String: " + length_str + " cm" + "<br />");
```

```
var num = parseInt(length_str) + 20;  
document.write("Length in Integer: " + num + " cm");
```

</script>

화면에 글 쓰기  
(JSP에서는 `out.print`,  
자바스크립트에서는 `document.write`)

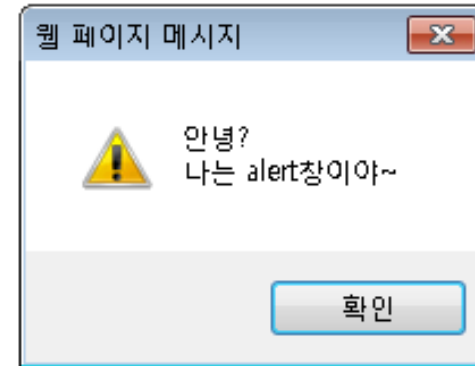




## 5. 팝업창 (3가지)

1

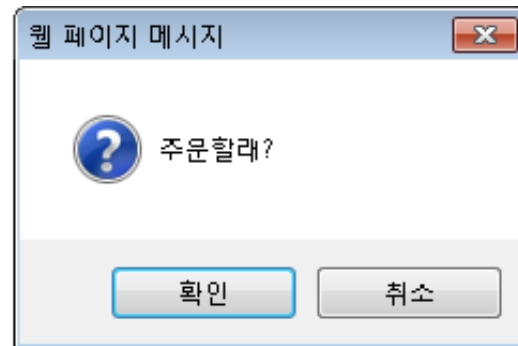
```
<script>  
    alert(" 안녕? \n\n 나는 alert창이야~");  
</script>
```



버튼 하나  
(리턴 없음)

2

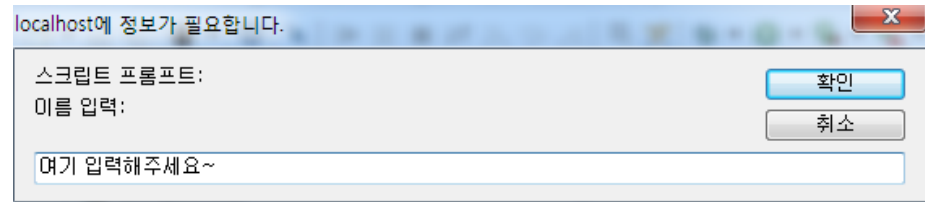
```
<script>  
    var answer = confirm("주문할래?");  
    document.write(answer + "를 선택하셨습니다 <br> 감사합니다~");  
</script>
```



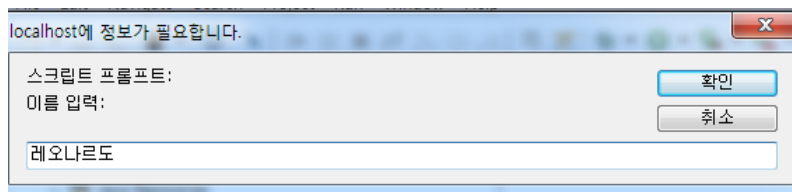
버튼 두 개 (true 혹은 false 리턴)

3

```
<script>  
var answer = prompt("이름 입력:", "여기 입력해주세요~");  
document.write("당신의 이름은? " + answer + "<br>");  
</script>
```



- 사용자로부터 키보드를 통해 문자열을 입력 받는다
- 대화상자 내에 메시지와 초기입력값이 입력 상자와 함께 표시됨
- “확인”을 누르면 입력된 문자열, “취소”를 누르면 **null**을 반환



당신의 이름은? 레오나르도

# 예제 (for문)

1~100까지의 합: 5050

```
<html>
  <head> <title> for문 </title> </head>
  <body>
    <script>
      var total = 0;
      for ( var n = 1 ; n <= 100 ; n++ )
      {
        total = total + n;
      }
      document.write("1~100까지의 합: ", total);
    </script>
  </body>
</html>
```

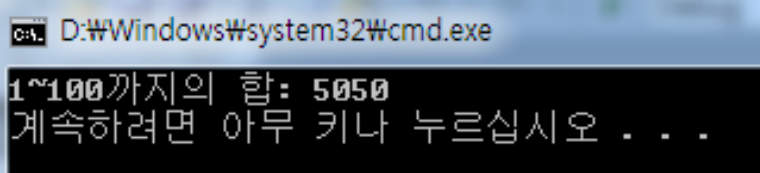
*var*  
안 써도  
돌아감

→

*document.write* 안에서는  
쉼표로 이어도 되고 +로 이어도 됨  
(단, +로 이르면, 숫자경우 숫자 더하기 값이 나올 수 있음)

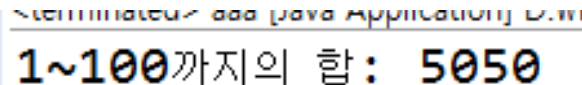
# C/C++, Java (for문)

```
#include <stdio.h>
int main() {
    int total = 0;
    for(int n=1; n<=100; n++ )
    {
        total = total + n;
    }
    printf("1~100까지의 합: %d \n", total);
}
```



A screenshot of a Windows command prompt window. The title bar shows the path "D:\Windows\system32\cmd.exe". The command prompt displays the output of a C program: "1~100까지의 합: 5050" followed by a prompt "계속하려면 아무 키나 누르십시오 . . .".

```
public class aaa {
    public static void main(String[] args) {
        int total = 0;
        for(int n=1; n<=100; n++ )
        {
            total = total + n;
        }
        System.out.println("1~100까지의 합: " + total);
    }
}
```



A screenshot of a Java application window titled "terminated: aaa [Java Application] D:\W...". The window displays the output of a Java program: "1~100까지의 합: 5050".

# 예제 (수식계산)

```
<html>
  <head> <title> 수식 계산하기 </title> </head>
  <body>
    <h2> 입력받은 두 수의 곱을 구하시오. </h2>
    <script>
      var a = prompt("첫 번째 수를 입력하세요!");
      var b = prompt("두 번째 수를 입력하세요!");
      var c = parseInt(a) * parseInt(b);

      document.write("결과: " + c);
    </script>
  </body>
</html>
```

자바스크립트 문법은

우리가 아는 C/C++/Java 와 상당히 비슷~

실습을 통해서 하나하나 배워보겠습니다.

# 실습1. 반지름 입력 받아서 원의 넓이 출력

<script>

```
var r = prompt("반지름 입력하세요.", "");
```

```
var area = parseInt(r) * parseInt(r) * 3.14;
```

```
document.write("반지름이 " + r + "인 원의 넓이: " + area);
```

</script>



반지름이 10.9인 원의 넓이: 314

정수로 바꿨으므로 반지름 10으로 계산됨

## 실습 2. 숫자 2개 입력 받아서 삼각형넓이, 사각형넓이, 사각형둘레 출력하기

```
<script>
```

```
var a = prompt("첫 번째 수를 입력하세요.", "");
```

```
var b = prompt("두 번째 수를 입력하세요.", "");
```

```
document.write("삼각형 넓이: " +  
                parseFloat(a)*parseFloat(b)/2 + "<br>");
```

```
document.write("사각형 넓이: " +  
                parseFloat(a)*parseFloat(b) + "<br>");
```

```
document.write("사각형 둘레: " +  
                (parseFloat(a)+parseFloat(b))*2 + "<br>");
```

```
</script>
```



실습3. 컨펌창 띄워서 “꽃무늬를 원해?” 물어본  
후, 확인누르면 화면이 온통 꽃. 취소누르  
면 온통 토끼

(1) html 하나로 코딩하는 방법

(2) html, jsp 2개로 코딩하는 방법

# html하나로 코딩하는 경우

1단계: 자바스크립트 없이 배경이 온통 꽃인 경우를 코딩해본다.

```
<head>                                참고: /5페이지처럼 인라인 스타일 코딩해도 됨
  <style>                                <body style = "background-image: url(flower.jpg)"></body>
    body {background-image: url(flower.jpg);}
  </style>
</head>
<body></body>
```

2단계: 자바스크립트(컨펌창) 추가해본다

```
<head>
  <script>
    var answer = confirm("꽃무늬를 원해?");
    if(answer)   <style> body{background-image: url(flower.jpg);} </style>
    else        <style> body{background-image: url(rabbit.jpg);} </style>
  </script>
</head>
<body></body>
```

3단계: 완성 (document.write는 html소스를 만드는 부분임)

```
<head>                                document.write는 html 코딩을 만들어주는 역할
  <script>
    var answer = confirm("꽃무늬를 원해?");
    if(answer)  document.write("<style> body{background-image: url(flower.jpg);} </style>");
    else        document.write("<style> body{background-image: url(rabbit.jpg);} </style>");
  </script>
</head>
<body></body>
```

# html, jsp 2개로 코딩하는 경우

a.html

```
<body>
  <script>
    var answer = confirm("꽃무늬를 원해?");
    if(answer)
      location.href="a.jsp?choice=flower";
    else
      location.href="a.jsp?choice=rabbit";
  </script>
</body>
```

a.jsp

```
<% String choice = request.getParameter("choice"); %>
<head>
  <style>
    body {background-image: url(<%=choice%>.jpg);}
  </style>
</head>
<body>
</body>
```

# 자바스크립트 함수

원의 넓이, 둘레

반지름

원의 넓이

원의 둘레

```
<script>
  function calculate(r)
  {
    naform.r1.value = r * r * Math.PI;    // 원의 넓이
    naform.r2.value = 2 * r * Math.PI;    // 원의 둘레
  }
</script>
```

타입 없이 바로 대개변수명

폼 안에 있는 인풋텍스트의 value

<h1> 원의 넓이와 둘레 계산 </h1>

<form name="naform">   
반지름 <input type="text" name="radius" size="10">

폼 안에 인풋텍스트 있으므로, 폼 이름도 주고 인풋텍스트 이름도 있어야

<input type="button" value="확인" onClick="calculate(radius.value)"> <br><br>

on으로 시작하는 이벤트 리스너들 중 하나

원의 넓이 <input type="text" name="r1" size="10"> <br><br>

원의 둘레 <input type="text" name="r2" size="10">

</form>

# 함수 사용해서 코딩해보기 (html 소스 1개로 완성)

## 1. 제곱 세제곱

(제곱함수 및 세제곱 함수 별도 사용)

숫자

제곱

세제곱



숫자

제곱

세제곱

## 2. 가로 세로 별사각형

가로

세로



```
*****  
*****  
*****  
*****  
*****  
*****  
*****
```

## 3. 구구단

이 페이지 내용: ×

구구단: 몇 단?



구구단  
[5단]

```
5 * 1 = 5  
5 * 2 = 10  
5 * 3 = 15  
5 * 4 = 20  
5 * 5 = 25  
5 * 6 = 30  
5 * 7 = 35  
5 * 8 = 40  
5 * 9 = 45
```

# HTML 태그의 이벤트 리스너 속성에 자바스크립트 코드 작성

```
<!DOCTYPE html>
<html>
<head>
<title>이벤트 리스너 속성에 자바스크립트 코드</title>
</head>
<body>
<h3>마우스 올려 보세요</h3>
<hr>

</body>
</html>
```

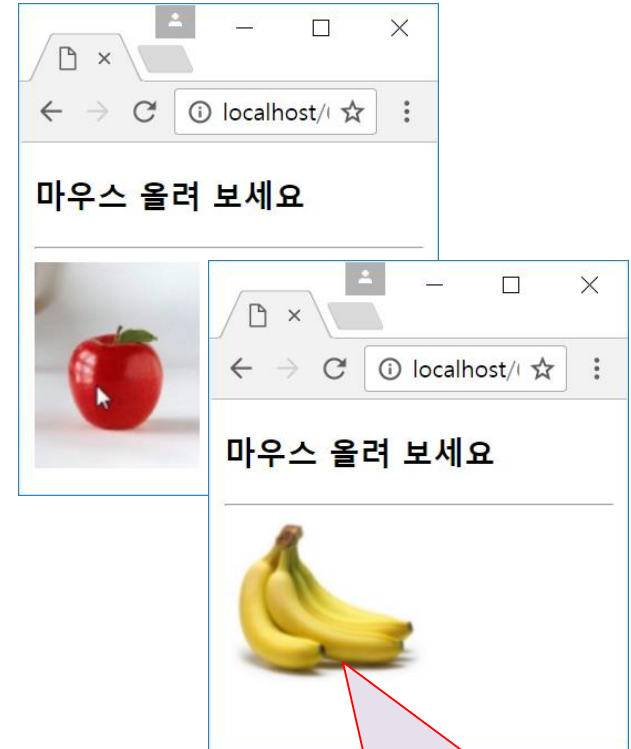
이벤트 리스너  
속성

this는 현재 img 태그를  
가리키는 자바스크립트 키워드

자바스크립트  
코드

onclick 생각나죠?

on으로 시작하는 이벤트 리스너들



이미지에 마우스를 올리면 바나나로  
내리면 다시 사과로 바뀐다.

# 입력자료 체크

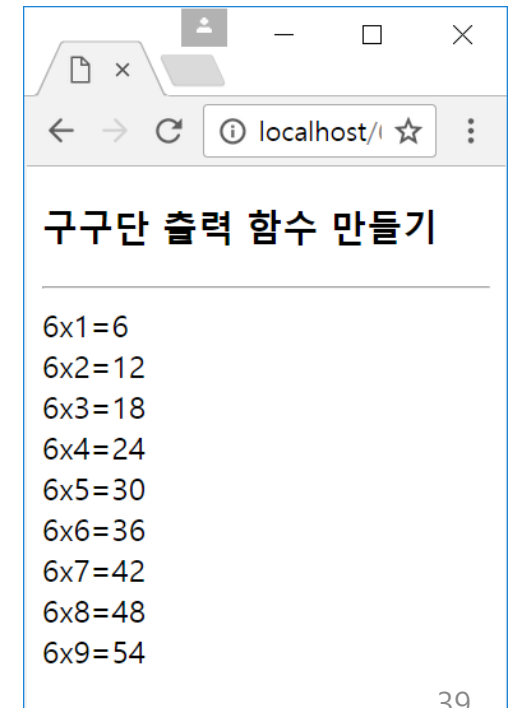
```
<!DOCTYPE html>
<html> <head> <title> 함수 만들기 </title>
<script>
function gugudan(n) { // 함수 작성
    var m = parseInt(n); // 문자열 n을 숫자로 바꿈
    if(isNaN(m) || m < 1 || m > 9) {
        alert("잘못 입력하셨습니다.");
        return;
    }
    for(var i=1; i<=9; i++) { // i는 1~9까지 반복
        document.write(m + "x" + i + "=" + m*i + "<br>");
    }
}
</script>
</head>
<body>
<h3>구구단 출력 함수 만들기</h3>
<hr>
<script>
    var n = prompt("구구단 몇 단을 원하세요", ""); // n은 문자열
    gugudan(n); // 함수 호출
</script>
</body>
</html>
```

n이 1~9사이의 숫자  
가 아닌 경우 처리

localhost 내용: ×

구구단 몇 단을 원하세요

확인 취소



JavaScript도 끝이 없음

진도는 여기까지~

남은 시간동안 계속 확인문제 코딩!